1 2 3 Ehrenfeucht: Descriptive Games

Neil Immerman

University of Massachusetts, Amherst, USA

`people.cs.umass.edu/~immerman`

Descriptive Complexity

Input $x_1\ x_2\ \cdots\ x_n$ $\mapsto$ $\boxed{\textbf{Computation}}$ $\mapsto$ **Output** $a_1\ a_2\ \cdots\ a_i\ \cdots\ a_m$

Descriptive Complexity

$$\begin{array}{ccccc} \textbf{Input} & & & & \textbf{Output} \\ x_1\ x_2\ \cdots\ x_n & \mapsto & \boxed{\textbf{Computation}} & \mapsto & a_1\ a_2\ \cdots\ \underset{S}{a_i}\ \cdots\ a_m \end{array}$$

Individual bits of the output are **decision problems.**

Descriptive Complexity

Input $\mapsto$ $\boxed{\textbf{Computation}}$ $\mapsto$ **Output**

$$x_1 \; x_2 \; \cdots \; x_n \quad \mapsto \quad \boxed{\textbf{Computation}} \quad \mapsto \quad a_1 \; a_2 \; \cdots \; \underset{S}{a_i} \; \cdots \; a_m$$

Individual bits of the output are **decision problems.**

Computational Complexity:
How hard is it to **check** if input has property S ?

Descriptive Complexity

Input $x_1 \; x_2 \; \cdots \; x_n$ $\mapsto$ $\boxed{\textbf{Computation}}$ $\mapsto$ **Output** $a_1 \; a_2 \; \cdots \; \underset{S}{a_i} \; \cdots \; a_m$

Individual bits of the output are **decision problems.**

Computational Complexity:
How hard is it to **check** if input has property S ?

Descriptive Complexity:
How rich a language do we need to **describe** property S?

Descriptive Complexity

Input **Output**

$$x_1\ x_2\ \cdots\ x_n \mapsto \boxed{\textbf{Computation}} \mapsto a_1\ a_2\ \cdots\ \underset{S}{a_i}\ \cdots\ a_m$$

Individual bits of the output are **decision problems.**

Computational Complexity:
How hard is it to **check** if input has property S ?

Descriptive Complexity:
How rich a language do we need to **describe** property S?

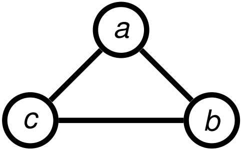
Constructive Isomorphism between these two approaches.

Input is Finite Ordered Structure

$$H = (\{a, b, c\}, \leq^H, E^H)$$

Graph $E^H = \{(a, b), (b, a), (b, c), (c, b), (c, a), (a, c)\}$

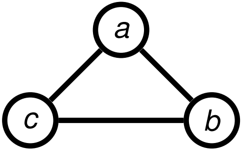
H



Input is Finite Ordered Structure

H		$=$	$(\{a, b, c\}, \leq^H, E^H)$
Ordered	$\leq^H$	$=$	$\{(a,a),(a,b),(a,c),(b,b),(b,c),(c,c)\}$
Graph	E^H	$=$	$\{(a,b),(b,a),(b,c),(c,b),(c,a),(a,c)\}$

H



First-Order Logic

input symbols:	$E, R, Y, B, \ldots$
variables:	$x, y, z, \ldots$
boolean connectives:	$\wedge, \vee, \neg$
quantifiers:	$\forall, \exists$
numeric symbols:	$=, \leq, min, max$

First-Order Logic

input symbols:	$E, R, Y, B, \ldots$
variables:	$x, y, z, \ldots$
boolean connectives:	$\wedge, \vee, \neg$
quantifiers:	$\forall, \exists$
numeric symbols:	$=, \leq, min, max$

In this setting, with the structure of interest being the **finite input**, FO is a **weak** complexity class.

First-Order Logic

input symbols: $E, R, Y, B, \ldots$
variables: $x, y, z, \ldots$
boolean connectives: $\wedge, \vee, \neg$
quantifiers: $\forall, \exists$
numeric symbols: $=, \leq, min, max$

In this setting, with the structure of interest being the **finite input**, FO is a **weak** complexity class.

It is **easy** to test if input, H, satisfies α $\quad (H \models \alpha)$.

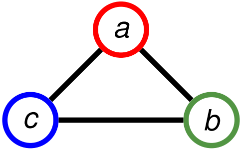
First-Order Logic

$H \qquad a \leq b \leq c$ $\qquad\qquad$ $G \qquad 1 \leq 2 \leq 3$

$$
\begin{aligned}
\alpha &\equiv \forall x \exists y\, E(x,y) \\
\beta &\equiv \forall xy\, (\neg E(x,x) \;\wedge\; (E(x,y) \rightarrow E(y,x))) \\
\gamma &\equiv \forall x\, ((\forall y\; x \leq y) \;\rightarrow\; R(x))
\end{aligned}
$$

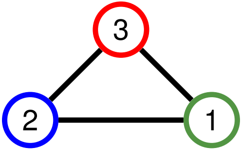
First-Order Logic

$H \qquad a \leq b \leq c$



$H \models \alpha \wedge \beta \wedge \gamma$

$G \qquad 1 \leq 2 \leq 3$



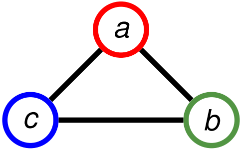
$$\alpha \equiv \forall x \exists y \, E(x, y)$$

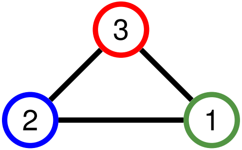
$$\beta \equiv \forall xy \, (\neg E(x, x) \ \wedge \ (E(x, y) \rightarrow E(y, x)))$$

$$\gamma \equiv \forall x \, ((\forall y \, x \leq y) \ \rightarrow \ R(x))$$

First-Order Logic

$H \quad a \leq b \leq c$ $\qquad\qquad$ $G \quad 1 \leq 2 \leq 3$





$H \models \alpha \wedge \beta \wedge \gamma$ $\qquad\qquad$ $G \models \alpha \wedge \beta \wedge \neg\gamma$

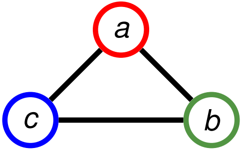
$$\alpha \equiv \forall x \exists y\, E(x, y)$$

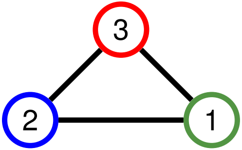
$$\beta \equiv \forall xy\, (\neg E(x, x) \;\wedge\; (E(x, y) \rightarrow E(y, x)))$$

$$\gamma \equiv \forall x\, ((\forall y\; x \leq y) \;\rightarrow\; R(x))$$

First-Order Logic

$H \quad a \le b \le c$ $\qquad$ $G \quad 1 \le 2 \le 3$





$H \models \alpha \wedge \beta \wedge \gamma$ $\qquad$ $G \models \alpha \wedge \beta \wedge \neg\gamma$

$$\alpha \equiv \forall x \exists y\, E(x,y)$$

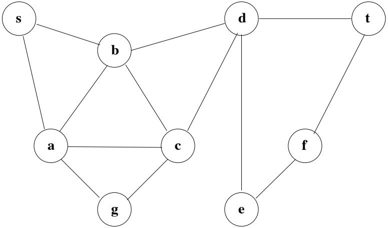
$$\beta \equiv \forall xy\, (\neg E(x,x) \ \wedge \ (E(x,y) \rightarrow E(y,x)))$$

$$\gamma \equiv \forall x\, ((\forall y\; x \le y) \ \rightarrow \ R(x))$$

α and β are **order independent**; γ is **order dependent**

Second-Order Logic: FO plus Relation Variables

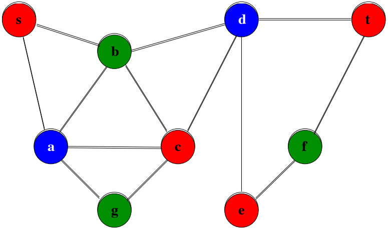
$$\Phi_{3\text{color}} \equiv \exists R^1\, G^1\, B^1\, \forall x\, y\, ((R(x) \lor G(x) \lor B(x)) \;\land\; (E(x,y) \to (\neg(R(x) \land R(y)) \land \neg(G(x) \land G(y)) \land \neg(B(x) \land B(y)))))$$



Second-Order Logic: FO plus Relation Variables

Fagin's Theorem: NP $=$ SO$\exists$

$$\Phi_{\text{3color}} \equiv \exists R^1 G^1 B^1 \forall x y ((R(x) \vee G(x) \vee B(x)) \wedge (E(x,y) \rightarrow (\neg(R(x) \wedge R(y)) \wedge \neg(G(x) \wedge G(y)) \wedge \neg(B(x) \wedge B(y)))))$$



Inductive Definition of Transitive Closure

$$R^\star(x, y) \quad \equiv \quad x = y \ \vee \ E(x, y) \ \vee \ \exists z(R^\star(x, z) \wedge R^\star(z, y))$$

Inductive Definition of Transitive Closure

$$R^\star(x,y) \;\; \equiv \;\; x = y \;\vee\; E(x,y) \;\vee\; \exists z(R^\star(x,z) \wedge R^\star(z,y))$$

$$\varphi_{tc}(R,x,y) \;\; \equiv \;\; x = y \;\vee\; E(x,y) \;\vee\; \exists z(R(x,z) \wedge R(z,y))$$

Inductive Definition of Transitive Closure

$$R^\star(x,y) \quad\equiv\quad x=y \ \vee\ E(x,y)\ \vee\ \exists z(R^\star(x,z)\wedge R^\star(z,y))$$

$$\varphi_{tc}(R,x,y) \quad\equiv\quad x=y \ \vee\ E(x,y)\ \vee\ \exists z(R(x,z)\wedge R(z,y))$$

$$\mathrm{LFP}(\varphi_{tc}) \quad=\quad \varphi_{tc}^{\lceil 1+\log n\rceil}(\emptyset) \quad=\quad R^\star$$

Inductive Definition of Transitive Closure

$$R^\star(x,y) \;\equiv\; x=y \;\vee\; E(x,y) \;\vee\; \exists z(R^\star(x,z)\wedge R^\star(z,y))$$

$$\varphi_{tc}(R,x,y) \;\equiv\; x=y \;\vee\; E(x,y) \;\vee\; \exists z(R(x,z)\wedge R(z,y))$$

$$\mathrm{LFP}(\varphi_{tc}) \;=\; \varphi_{tc}^{\lceil 1+\log n\rceil}(\emptyset) \;=\; R^\star$$

Next, we'll sketch that every first-order relational operator such as φ_{tc} is equivalent to a block of restricted quantifiers. Thus the LFP is just the iteration of a quantifier block.

$\varphi_{tc}(R,x,y) \;\equiv\; x = y \lor E(x,y) \lor \exists z\,(R(x,z) \land R(z,y))$

1. Dummy universal quantification for base case:

$$\begin{aligned}\varphi_{tc}(R,x,y) &\equiv (\forall z.M_1)(\exists z)(R(x,z) \land R(z,y)) \\ M_1 &\equiv \neg(x = y \lor E(x,y))\end{aligned}$$

$\varphi_{tc}(R,x,y) \;\equiv\; x = y \vee E(x,y) \vee \exists z\,(R(x,z) \wedge R(z,y))$

1. Dummy universal quantification for base case:

$$\begin{aligned}\varphi_{tc}(R,x,y) &\equiv (\forall z.M_1)(\exists z)(R(x,z) \wedge R(z,y))\\ M_1 &\equiv \neg(x = y \vee E(x,y))\end{aligned}$$

2. Using $\forall$, replace two occurrences of R with one:

$$\begin{aligned}\varphi_{tc}(R,x,y) &\equiv (\forall z.M_1)(\exists z)(\forall uv.M_2)R(u,v)\\ M_2 &\equiv (u = x \wedge v = z) \;\vee\; (u = z \wedge v = y)\end{aligned}$$

$\varphi_{tc}(R,x,y) \;\equiv\; x = y \vee E(x,y) \vee \exists z\,(R(x,z) \wedge R(z,y))$

1. Dummy universal quantification for base case:

$$\begin{aligned}\varphi_{tc}(R,x,y) &\equiv (\forall z.M_1)(\exists z)(R(x,z) \wedge R(z,y))\\ M_1 &\equiv \neg(x = y \vee E(x,y))\end{aligned}$$

2. Using $\forall$, replace two occurrences of R with one:

$$\begin{aligned}\varphi_{tc}(R,x,y) &\equiv (\forall z.M_1)(\exists z)(\forall uv.M_2)R(u,v)\\ M_2 &\equiv (u = x \wedge v = z) \;\vee\; (u = z \wedge v = y)\end{aligned}$$

3. Requantify x and y.

$$M_3 \equiv (x = u \wedge y = v)$$

$$\varphi_{tc}(R,x,y) \equiv [\,(\forall z.M_1)(\exists z)(\forall uv.M_2)(\exists xy.M_3)\,]\,R(x,y)$$

Every FO inductive definition is equivalent to a quantifier block.

$$\mathrm{CRAM}[t(n)] \quad = \quad \text{concurrent parallel random access machine; polynomial hardware, parallel time } O(t(n))$$

$$\mathrm{IND}[t(n)] \quad = \quad \text{first-order, depth } t(n) \text{ inductive definitions}$$

$$\mathrm{FO}[t(n)] \quad = \quad t(n) \text{ repetitions of a block of restricted quantifiers:}$$

$$\mathrm{QB} \quad = \quad [(Q_1 x_1 . M_1) \cdots (Q_k x_k . M_k)]; \quad M_i \text{ quantifier-free}$$

$$\varphi_n \quad = \quad \underbrace{[\mathrm{QB}][\mathrm{QB}] \cdots [\mathrm{QB}]}_{t(n)} M_0$$

parallel time = inductive depth = QB iteration

Thm. For all constructible, polynomially bounded $t(n)$,

$$\mathrm{CRAM}[t(n)] \;=\; \mathrm{IND}[t(n)] \;=\; \mathrm{FO}[t(n)]$$

parallel time = inductive depth = QB iteration

Thm. For all constructible, polynomially bounded $t(n)$,

$$\mathrm{CRAM}[t(n)] \;=\; \mathrm{IND}[t(n)] \;=\; \mathrm{FO}[t(n)]$$

Thm. For all $t(n)$, even beyond polynomial,

$$\mathrm{CRAM}[t(n)] \quad = \quad \mathrm{FO}[t(n)]$$

parallel time = inductive depth = QB iteration

Thm. For all constructible, polynomially bounded $t(n)$,

$$\mathrm{CRAM}[t(n)] \;=\; \mathrm{IND}[t(n)] \;=\; \mathrm{FO}[t(n)]$$

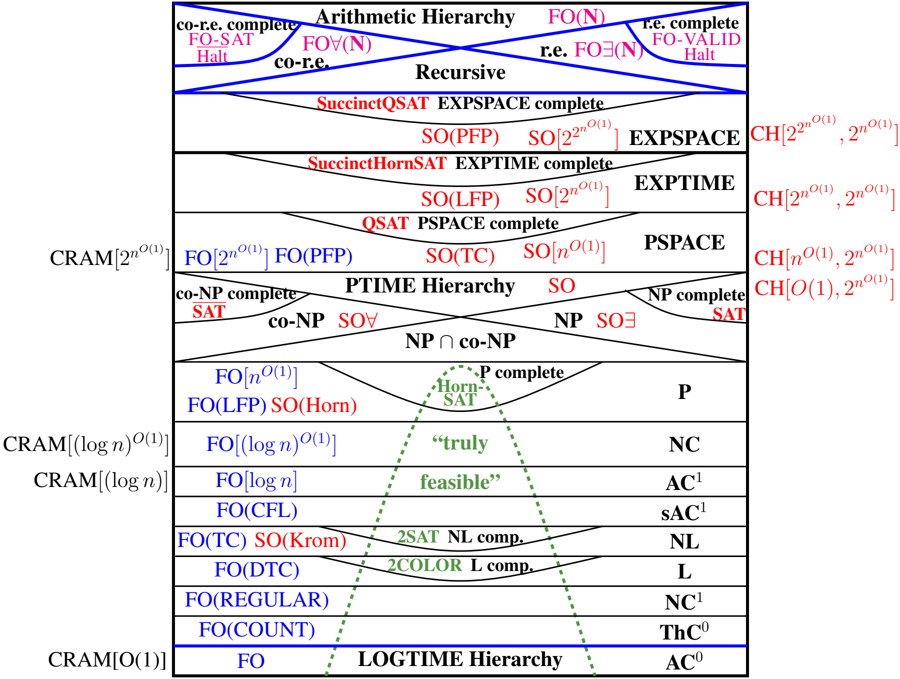
Thm. For all $t(n)$, even beyond polynomial,

$$\mathrm{CRAM}[t(n)] \quad = \quad \mathrm{FO}[t(n)]$$

Thm. For all $t(n)$,

$$\mathrm{CH}[t(n), 2^{n^{O(1)}}] \quad = \quad \mathrm{SO}[t(n)]$$

$\mathrm{CH}[t(n), h(n)]$ is parallel time $O(t(n))$ on a CRAM with $O(h(n))$ hardware gates.



Arithmetic Hierarchy FO(**N**)

co-r.e. complete
FO-SAT
$\overline{\text{Halt}}$

FO∀(**N**)

co-r.e.

r.e. complete
FO-VALID
Halt

r.e. FO∃(**N**)

Recursive

SuccinctQSAT **EXPSPACE complete**

SO(PFP) SO[$2^{2^{n^{O(1)}}}$] **EXPSPACE** CH[$2^{2^{n^{O(1)}}}, 2^{n^{O(1)}}$]

SuccinctHornSAT **EXPTIME complete**

EXPTIME

SO(LFP) SO[$2^{n^{O(1)}}$] CH[$2^{n^{O(1)}}, 2^{n^{O(1)}}$]

QSAT **PSPACE complete**

CRAM[$2^{n^{O(1)}}$] FO[$2^{n^{O(1)}}$] FO(PFP) SO(TC) SO[$n^{O(1)}$] **PSPACE** CH[$n^{O(1)}, 2^{n^{O(1)}}$]

PTIME Hierarchy SO CH[$O(1), 2^{n^{O(1)}}$]

co-NP complete
$\overline{\textbf{SAT}}$

co-NP SO∀

NP SO∃

NP complete
SAT

NP ∩ co-NP

FO[$n^{O(1)}$] **P complete**

Horn-SAT

FO(LFP) SO(Horn) **P**

CRAM[$(\log n)^{O(1)}$] FO[$(\log n)^{O(1)}$] **“truly** **NC**

CRAM[$(\log n)$] FO[$\log n$] **feasible”** **AC**^{1}

FO(CFL) **sAC**^{1}

FO(TC) SO(Krom) **2SAT** **NL comp.** **NL**

FO(DTC) **2COLOR** **L comp.** **L**

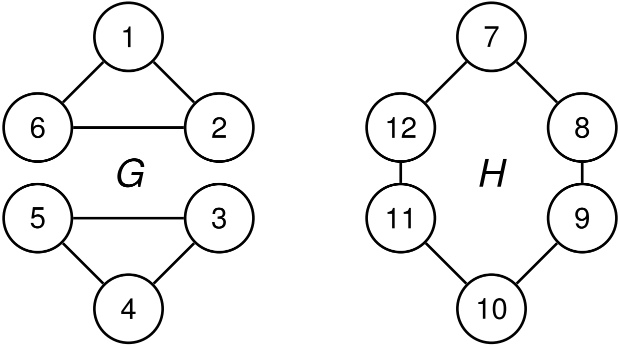
FO(REGULAR) **NC**^{1}

FO(COUNT) **ThC**^{0}

CRAM[O(1)] FO **LOGTIME Hierarchy** **AC**^{0}

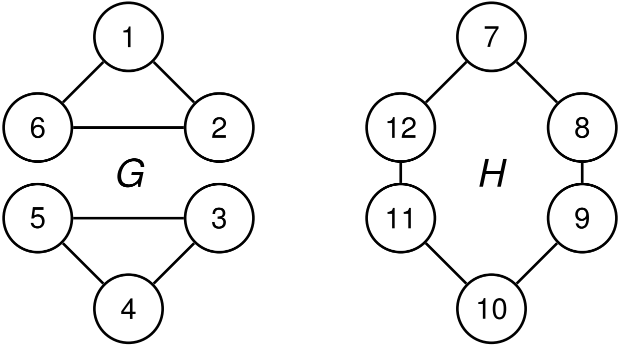
Ehrenfeucht-Fraïssé Game

$\mathcal{G}_m^c(G, H)$ m moves, c colors,



Ehrenfeucht-Fraïssé Game

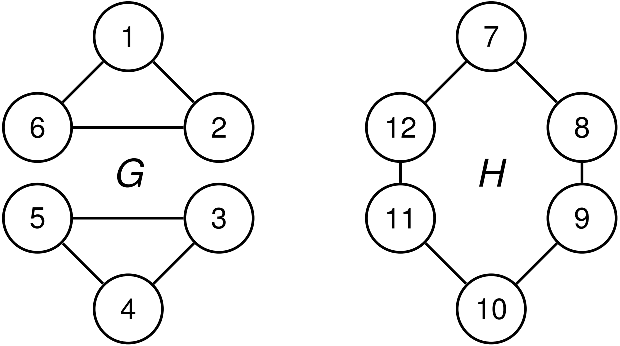
$\mathcal{G}_m^c(G, H)$ m moves, c colors, **Spoiler**: show difference



Ehrenfeucht-Fraïssé Game

$\mathcal{G}_m^c(G,H)$ m moves, c colors, **Spoiler**: show difference

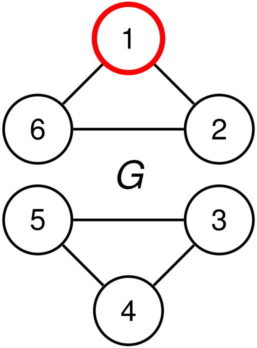
Duplicator: preserve isomorphism of induced substructures

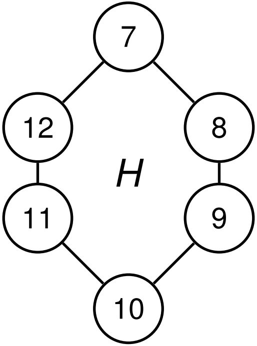


Ehrenfeucht-Fraïssé Game

$\mathcal{G}^c_m(G, H)$ m moves, c colors, **Spoiler**: show difference

Duplicator: preserve isomorphism of induced substructures

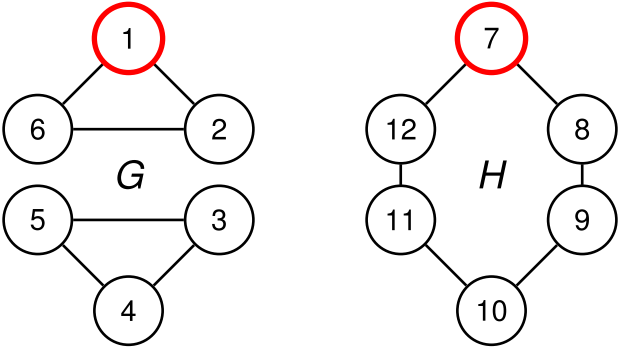




Ehrenfeucht-Fraïssé Game

$\mathcal{G}_m^c(G, H)$ m moves, c colors, **Spoiler**: show difference

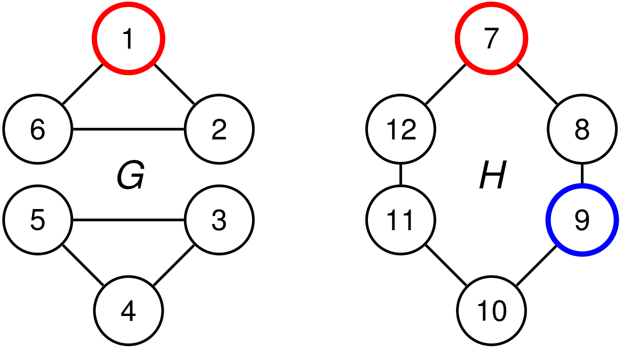
Duplicator: preserve isomorphism of induced substructures



Ehrenfeucht-Fraïssé Game

$\mathcal{G}^c_m(G, H)$ m moves, c colors, **Spoiler**: show difference

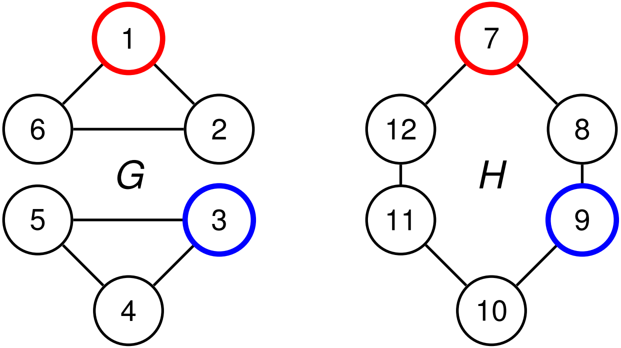
Duplicator: preserve isomorphism of induced substructures



Ehrenfeucht-Fraïssé Game

$\mathcal{G}_m^c(G,H)$ m moves, c colors, **Spoiler**: show difference

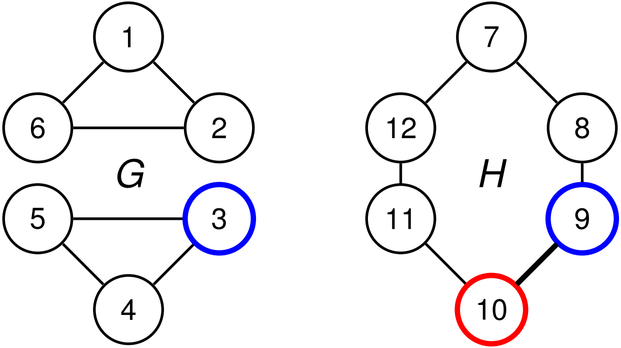
Duplicator: preserve isomorphism of induced substructures



Ehrenfeucht-Fraïssé Game

$\mathcal{G}_m^c(G, H)$ m moves, c colors, **Spoiler**: show difference

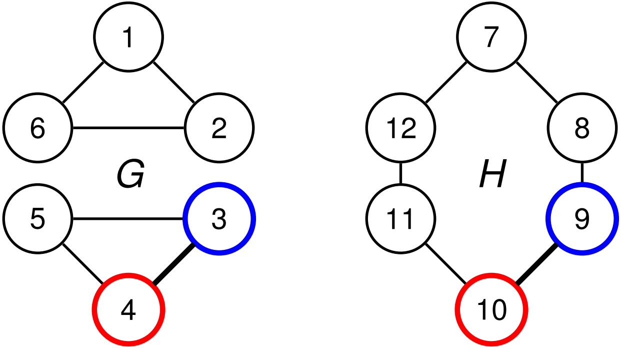
Duplicator: preserve isomorphism of induced substructures



Ehrenfeucht-Fraïssé Game

$\mathcal{G}^c_m(G, H)$ m moves, c colors, **Spoiler**: show difference

Duplicator: preserve isomorphism of induced substructures

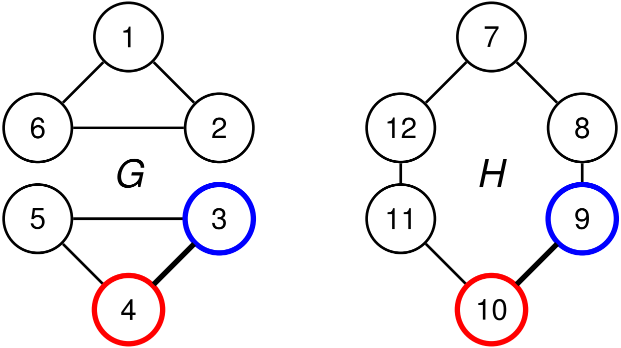


Ehrenfeucht-Fraïssé Game

$\mathcal{G}^c_m(G, H)$ m moves, c colors, **Spoiler**: show difference

Duplicator: preserve isomorphism of induced substructures

For all m, **D** wins $\mathcal{G}^2_m(G, H)$;

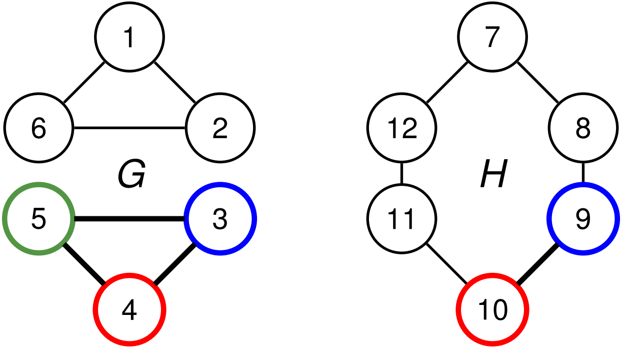


Ehrenfeucht-Fraïssé Game

$\mathcal{G}_m^c(G, H)$ m moves, c colors,   **Spoiler**: show difference

Duplicator: preserve isomorphism of induced substructures

For all m, **D** wins $\mathcal{G}_m^2(G, H)$;

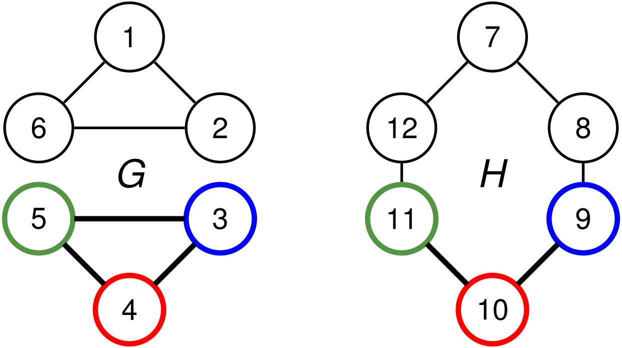


Ehrenfeucht-Fraïssé Game

$\mathcal{G}_m^c(G, H)$ m moves, c colors, **Spoiler**: show difference

Duplicator: preserve isomorphism of induced substructures

For all m, **D** wins $\mathcal{G}_m^2(G, H)$;

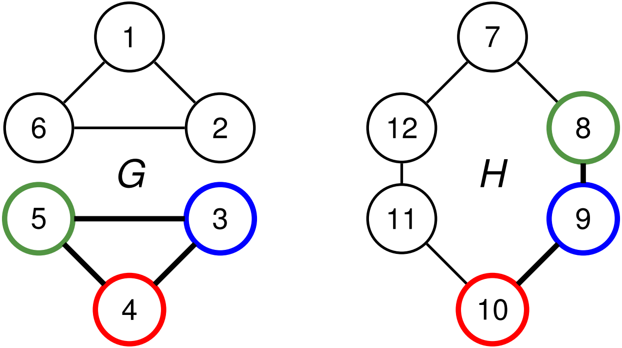


Ehrenfeucht-Fraïssé Game

$\mathcal{G}_m^c(G,H)$ m moves, c colors, **Spoiler**: show difference

Duplicator: preserve isomorphism of induced substructures

For all m, **D** wins $\mathcal{G}_m^2(G,H)$;



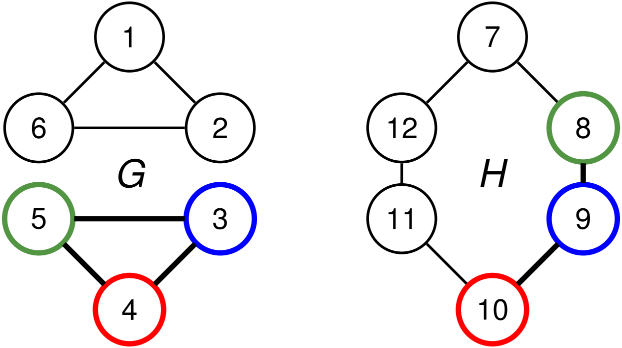
Ehrenfeucht-Fraïssé Game

$\mathcal{G}_m^c(G,H)$ m moves, c colors, **Spoiler**: show difference

Duplicator: preserve isomorphism of induced substructures

For all m, **D** wins $\mathcal{G}_m^2(G,H)$; but **S** wins $\mathcal{G}_3^3(G,H)$.

$$\varphi \;\equiv\; \exists rbg(E(r,b) \wedge E(b,g) \wedge E(g,r)) \qquad G \models \varphi; \quad H \models \neg\varphi$$



Fundamental Thm of Ehrenfeucht-Fraïssé Games

Notation: $G \sim_m^c H$ means that **Duplicator** has a winning strategy for $\mathcal{G}_m^c(G, H)$.

Fundamental Thm of Ehrenfeucht-Fraïssé Games

Notation: $G \sim^c_m H$ means that **Duplicator** has a winning strategy for $\mathcal{G}^c_m(G, H)$.

Thm. **D** has a winning strategy on the *m*-move, *c*-color game on G, H iff G and H agree on all formulas using c variables and quantifier depth m,

$$G \sim^c_m H \quad \Leftrightarrow \quad G \equiv^c_m H$$

Fundamental Thm of Ehrenfeucht-Fraïssé Games

Notation: $G \sim_m^c H$ means that **Duplicator** has a winning strategy for $\mathcal{G}_m^c(G, H)$.

Thm. **D** has a winning strategy on the *m*-move, *c*-color game on G, H iff G and H agree on all formulas using c variables and quantifier depth m,

$$G \sim_m^c H \quad \Leftrightarrow \quad G \equiv_m^c H$$

Ehrenfeucht-Fraïssé games are **fantastically useful** for determining what is expressible in FO logic in a given quantifier depth and with a given number of variables.

Fundamental Thm of Ehrenfeucht-Fraïssé Games

Notation: $G \sim_m^c H$ means that **Duplicator** has a winning strategy for $\mathcal{G}_m^c(G, H)$.

Thm. **D** has a winning strategy on the *m*-move, *c*-color game on G, H iff G and H agree on all formulas using *c* variables and quantifier depth *m*,

$$G \sim_m^c H \quad \Leftrightarrow \quad G \equiv_m^c H$$

Ehrenfeucht-Fraïssé games are **fantastically useful** for determining what is expressible in FO logic in a given quantifier depth and with a given number of variables.

But, as we will see next, Ehrenfeucht-Fraïssé games are **not very helpful** for proving Descriptive Lower Bounds.

Thm. $\mathcal{L}^{3}_{\lceil 2+\log n\rceil}$ suffices to characterize **any property whatsoever** over ordered graphs.

Thm. $\mathcal{L}^3_{\lceil 2+\log n\rceil}$ suffices to characterize **any property whatsoever** over ordered graphs.

Proof: We can name any vertex by number in $\mathcal{L}^3_{\lceil 1+\log n\rceil}$.

Thm. $\mathcal{L}^3_{\lceil 2+\log n\rceil}$ suffices to characterize **any property whatsoever** over ordered graphs.

Proof: We can name any vertex by number in $\mathcal{L}^3_{\lceil 1+\log n\rceil}$.

We can identify a graph in $\mathcal{L}^3_{\lceil 2+\log n\rceil}$ by asserting for each $i, j \leq n$, whether $E(v_i, v_j)$.

Thm. $\mathcal{L}^3_{\lceil 2+\log n\rceil}$ suffices to characterize **any property whatsoever** over ordered graphs.

Proof: We can name any vertex by number in $\mathcal{L}^3_{\lceil 1+\log n\rceil}$.

We can identify a graph in $\mathcal{L}^3_{\lceil 2+\log n\rceil}$ by asserting for each $i, j \leq n$, whether $E(v_i, v_j)$.

In $\mathcal{L}^3_{\lceil 2+\log n\rceil}$, we can identify an arbitrary set of graphs on n vertices. □

Number of Quantifiers game:

- Separation Game: [I81]

Number of Quantifiers game:

- Separation Game: [I81]
- renamed Multistructural Game: [FLRV21]: LICS21, determined exact number of quantifiers to identify a linear order of length n.

Number of Quantifiers game:

- Separation Game: [I81]
- renamed Multistructural Game: [FLRV21]: LICS21, determined exact number of quantifiers to identify a linear order of length *n*.
- [FLVW22]

Number of Quantifiers game:

- Separation Game: [I81]
- renamed Multistructural Game: [FLRV21]: LICS21, determined exact number of quantifiers to identify a linear order of length n.
- [FLVW22]
- [CFIKLS23] – next talk by Rik Sengupta

Number of Quantifiers game:

- Separation Game: [I81]
- renamed Multistructural Game: [FLRV21]: LICS21, determined exact number of quantifiers to identify a linear order of length *n*.
- [FLVW22]
- [CFIKLS23] – next talk by Rik Sengupta

Personal history of my 1980 Ph.D. thesis:

Number of Quantifiers game:

- Separation Game: [I81]
- renamed Multistructural Game: [FLRV21]: LICS21, determined exact number of quantifiers to identify a linear order of length *n*.
- [FLVW22]
- [CFIKLS23] – next talk by Rik Sengupta

Personal history of my 1980 Ph.D. thesis:

- 1978: Larry Carter sends me via snail mail a hard copy of R. Fagin, “Generalized First-Order Spectra and Polynomial-Time Recognizable Sets.”

Number of Quantifiers game:

- Separation Game: [I81]
- renamed Multistructural Game: [FLRV21]: LICS21, determined exact number of quantifiers to identify a linear order of length *n*.
- [FLVW22]
- [CFIKLS23] – next talk by Rik Sengupta

Personal history of my 1980 Ph.D. thesis:

- 1978: Larry Carter sends me via snail mail a hard copy of R. Fagin, "Generalized First-Order Spectra and Polynomial-Time Recognizable Sets."
- "Number of Quantifiers is Better Than Number of Tape Cells," JCSS (1981), prelim. version: "Length of Predicate Calculus Formulas as a New Complexity Measure," FOCS (1979).

Number of Quantifiers game:

- Separation Game: [I81]
- renamed Multistructural Game: [FLRV21]: LICS21, determined exact number of quantifiers to identify a linear order of length *n*.
- [FLVW22]
- [CFIKLS23] – next talk by Rik Sengupta

Personal history of my 1980 Ph.D. thesis:

- 1978: Larry Carter sends me via snail mail a hard copy of R. Fagin, “Generalized First-Order Spectra and Polynomial-Time Recognizable Sets.”
- “Number of Quantifiers is Better Than Number of Tape Cells,” JCSS (1981), prelim. version: “Length of Predicate Calculus Formulas as a New Complexity Measure,” FOCS (1979).
- “Upper and Lower Bounds for First Order Expressibility,” JCSS (1982), prelim. version: FOCS (1980).

Number of Quantifiers game = Multistructural Game

$MS_m(\mathcal{A}, \mathcal{B})$; m moves played on a pair of sets of structures.

Number of Quantifiers game = Multistructural Game

$MS_m(\mathcal{A}, \mathcal{B})$; m moves played on a pair of sets of structures.

Spoiler chooses an element of each structure on one side.

Number of Quantifiers game = Multistructural Game

$MS_m(\mathcal{A}, \mathcal{B})$; m moves played on a pair of sets of structures.

Spoiler chooses an element of each structure on one side.

Duplicator makes multiple copies of each structure on the other side and then chooses a corresponding element of each structure.

Number of Quantifiers game = Multistructural Game

$MS_m(\mathcal{A}, \mathcal{B})$; *m* moves played on a pair of sets of structures.

Spoiler chooses an element of each structure on one side.

Duplicator makes multiple copies of each structure on the other side and then chooses a corresponding element of each structure.

Duplicator wins if after each move there is a a pair of isomorphic induced substructures, one from each side.

Number of Quantifiers game = Multistructural Game

$MS_m(\mathcal{A}, \mathcal{B})$; m moves played on a pair of sets of structures.

Spoiler chooses an element of each structure on one side.

Duplicator makes multiple copies of each structure on the other side and then chooses a corresponding element of each structure.

Duplicator wins if after each move there is a a pair of isomorphic induced substructures, one from each side.

Thm. Spoiler wins $MS_m(\mathcal{A}, \mathcal{B})$ iff there is a formula φ having at most m quantifiers, $\mathcal{A} \models \varphi$; $\mathcal{B} \models \neg\varphi$.

Number of Quantifiers game = Multistructural Game

$MS_m(\mathcal{A}, \mathcal{B})$; m moves played on a pair of sets of structures.

Spoiler chooses an element of each structure on one side.

Duplicator makes multiple copies of each structure on the other side and then chooses a corresponding element of each structure.

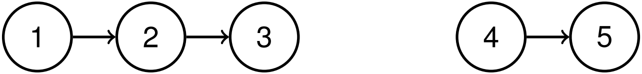
Duplicator wins if after each move there is a a pair of isomorphic induced substructures, one from each side.

Thm. Spoiler wins $MS_m(\mathcal{A}, \mathcal{B})$ iff there is a formula φ having at most m quantifiers, $\quad \mathcal{A} \models \varphi; \quad \mathcal{B} \models \neg\varphi$.

Cor. Property S is expressible with $m(n)$ quantifiers, for inputs of size n iff Spoiler wins $MS_m(S_n, \overline{S}_n)$ where S_n is the set of all ordered structures of size n satisfying S and $\overline{S}_n$ is the set of all ordered structures of size n not satisfying S.

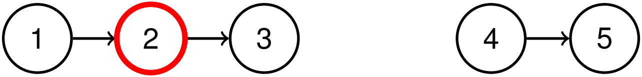
Examples: $MS_2(\mathcal{A}, \mathcal{B})$ and $MS_3(\mathcal{A}, \mathcal{B})$ Games

$\mathcal{A} = \{L_3\}$ $\mathcal{B} = \{L_2\}$



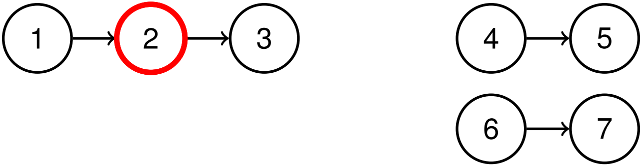
Examples: $MS_2(\mathcal{A}, \mathcal{B})$ and $MS_3(\mathcal{A}, \mathcal{B})$ Games

$\mathcal{A} = \{L_3\}$ $\qquad$ $\mathcal{B} = \{L_2\}$



Examples: $MS_2(\mathcal{A}, \mathcal{B})$ and $MS_3(\mathcal{A}, \mathcal{B})$ Games

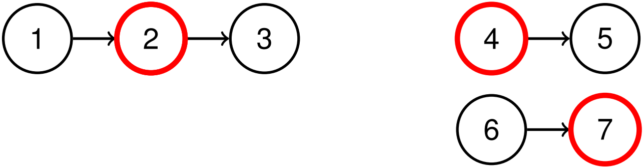
$\mathcal{A} = \{L_3\}$ $\qquad$ $\mathcal{B} = \{L_2\}$



Examples: $MS_2(\mathcal{A}, \mathcal{B})$ and $MS_3(\mathcal{A}, \mathcal{B})$ Games

$$\mathcal{A} = \{L_3\} \qquad \mathcal{B} = \{L_2\}$$

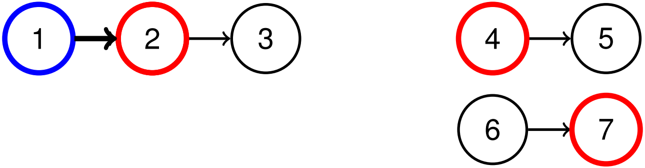
$$\mathcal{A}_1 = \{(L_3, 2)\} \qquad \mathcal{B}_1 = \{(L_2, 4), (L_2', 7)\}$$



Examples: $MS_2(\mathcal{A}, \mathcal{B})$ and $MS_3(\mathcal{A}, \mathcal{B})$ Games

$\mathcal{A} = \{L_3\}$ $\qquad$ $\mathcal{B} = \{L_2\}$

$\mathcal{A}_1 = \{(L_3, 2)\}$ $\qquad$ $\mathcal{B}_1 = \{(L_2, 4), (L_2', 7)\}$



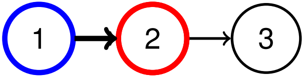
Examples: $MS_2(\mathcal{A},\mathcal{B})$ and $MS_3(\mathcal{A},\mathcal{B})$ Games

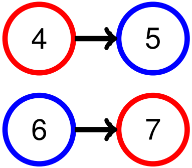
Duplicator wins $MS_2(\mathcal{A},\mathcal{B})$

$$\mathcal{A} = \{L_3\} \qquad \mathcal{B} = \{L_2\}$$

$$\mathcal{A}_1 = \{(L_3, 2)\} \qquad \mathcal{B}_1 = \{(L_2, 4), (L_2', 7)\}$$

$$\mathcal{A}_2 = \{(L_3, 2, 1)\} \qquad \mathcal{B}_2 = \{(L_2, 4, 5), (L_2', 7, 6)\}$$





Examples: $MS_2(\mathcal{A}, \mathcal{B})$ and $MS_3(\mathcal{A}, \mathcal{B})$ Games

Duplicator wins $MS_2(\mathcal{A}, \mathcal{B})$ Spoiler wins $MS_3(\mathcal{A}, \mathcal{B})$

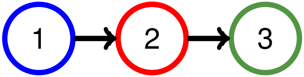
$$\begin{array}{ll} \mathcal{A} = \{L_3\} & \mathcal{B} = \{L_2\} \\ \mathcal{A}_1 = \{(L_3, 2)\} & \mathcal{B}_1 = \{(L_2, 4), (L_2', 7)\} \\ \mathcal{A}_2 = \{(L_3, 2, 1)\} & \mathcal{B}_2 = \{(L_2, 4, 5), (L_2', 7, 6)\} \end{array}$$

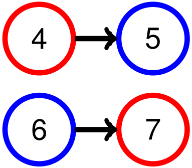
Examples: $MS_2(\mathcal{A}, \mathcal{B})$ and $MS_3(\mathcal{A}, \mathcal{B})$ Games

Duplicator wins $MS_2(\mathcal{A}, \mathcal{B})$ Spoiler wins $MS_3(\mathcal{A}, \mathcal{B})$

$$\begin{array}{ll} \mathcal{A} = \{L_3\} & \mathcal{B} = \{L_2\} \\ \mathcal{A}_1 = \{(L_3, 2)\} & \mathcal{B}_1 = \{(L_2, 4), (L_2', 7)\} \\ \mathcal{A}_2 = \{(L_3, 2, 1)\} & \mathcal{B}_2 = \{(L_2, 4, 5), (L_2', 7, 6)\} \end{array}$$

$$\varphi \equiv \exists rbg(E(b, r) \wedge E(r, g)) \qquad \mathcal{A} \models \varphi \quad \mathcal{B} \models \neg\varphi$$





Examples: $MS_2(\mathcal{A},\mathcal{B})$ and $MS_3(\mathcal{A},\mathcal{B})$ Games

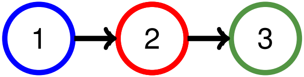
Duplicator wins $MS_2(\mathcal{A},\mathcal{B})$ Spoiler wins $MS_3(\mathcal{A},\mathcal{B})$

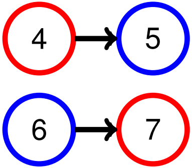
Spoiler wins $\mathcal{G}_2^2(L_3, L_2)$

$$\begin{array}{ll} \mathcal{A} = \{L_3\} & \mathcal{B} = \{L_2\} \\ \mathcal{A}_1 = \{(L_3, 2)\} & \mathcal{B}_1 = \{(L_2, 4), (L_2', 7)\} \\ \mathcal{A}_2 = \{(L_3, 2, 1)\} & \mathcal{B}_2 = \{(L_2, 4, 5), (L_2', 7, 6)\} \end{array}$$

$$\varphi \;\equiv\; \exists rbg(E(b, r) \wedge E(r, g)) \qquad \mathcal{A} \models \varphi \quad \mathcal{B} \models \neg\varphi$$

$$\psi \;\equiv\; \exists r(\exists b(E(b, r)) \;\wedge\; \exists b E(r, b)) \quad \mathcal{A} \models \varphi \quad \mathcal{B} \models \neg\varphi$$





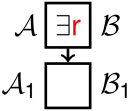
Size Game:[AI03]; QVT: [CFIKLS23]

$QVT^{c}_{m}(\mathcal{A}, \mathcal{B})$ Spoiler builds formula tree separating $\mathcal{A}$, $\mathcal{B}$.

$$\mathcal{A}\ \square\ \mathcal{B}$$

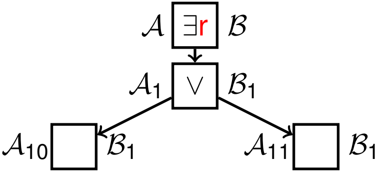
Size Game:[AI03]; QVT: [CFIKLS23]

$QVT_m^c(\mathcal{A}, \mathcal{B})$ Spoiler builds formula tree separating $\mathcal{A}$, $\mathcal{B}$.



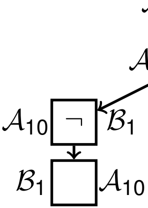
Size Game:[AI03]; QVT: [CFIKLS23]

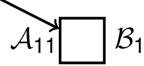
$QVT_m^c(\mathcal{A}, \mathcal{B})$ Spoiler builds formula tree separating $\mathcal{A}$, $\mathcal{B}$.



Size Game:[AI03]; QVT: [CFIKLS23]

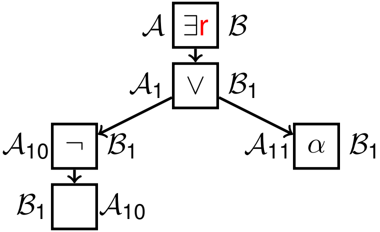
$QVT_m^c(\mathcal{A}, \mathcal{B})$ Spoiler builds formula tree separating $\mathcal{A}$, $\mathcal{B}$.





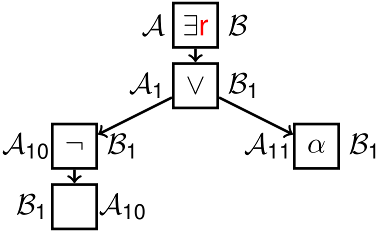
Size Game:[AI03]; QVT: [CFIKLS23]

$QVT^{c}_{m}(\mathcal{A},\mathcal{B})$ Spoiler builds formula tree separating $\mathcal{A}$, $\mathcal{B}$.



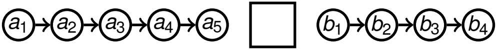
Size Game:[AI03]; QVT: [CFIKLS23]

$QVT_m^c(\mathcal{A},\mathcal{B})$ Spoiler builds formula tree separating $\mathcal{A}$, $\mathcal{B}$.

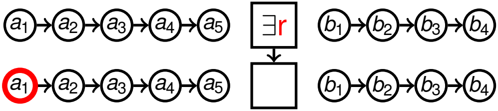


Thm. Spoiler can close the $QVT_m^c(\mathcal{A},\mathcal{B})$ game tree using c colors and m quantifier moves iff there is a formula with c variables and m quantifiers separating $\mathcal{A}$ from $\mathcal{B}$.

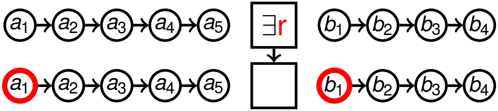
$QVT^2(L_5, L_4)$



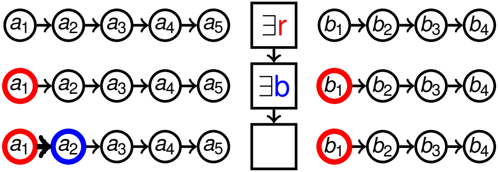
$QVT^2(L_5, L_4)$



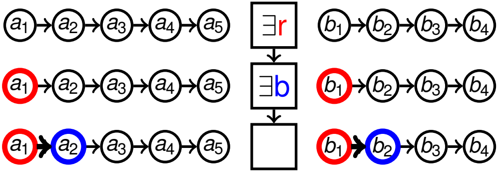
$QVT^2(L_5, L_4)$



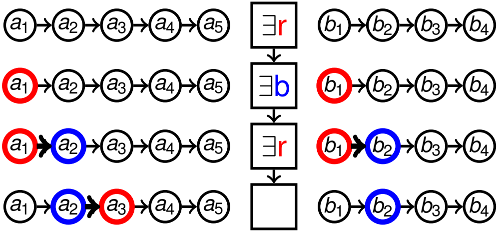
$QVT^2(L_5, L_4)$



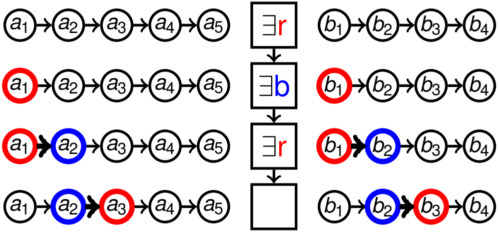
$QVT^2(L_5, L_4)$



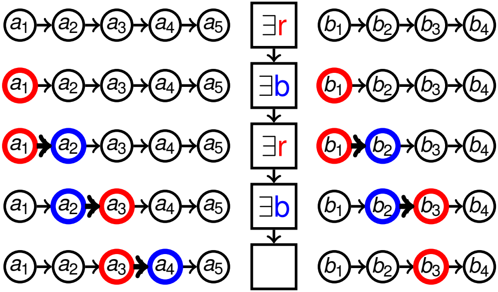
$QVT^2(L_5, L_4)$



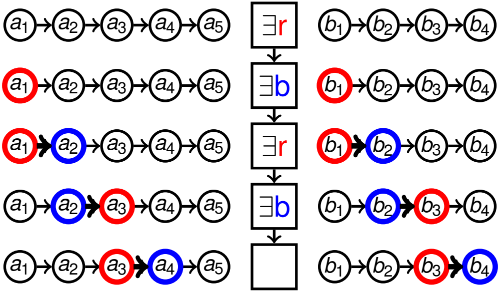
$QVT^2(L_5, L_4)$



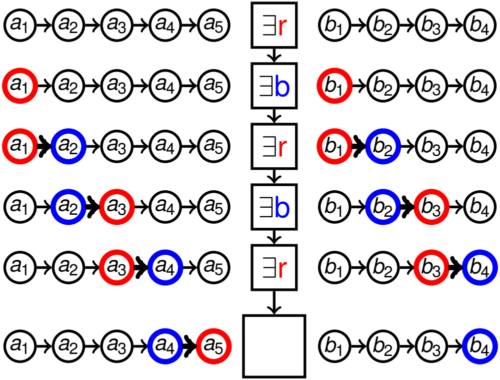
$QVT^2(L_5, L_4)$



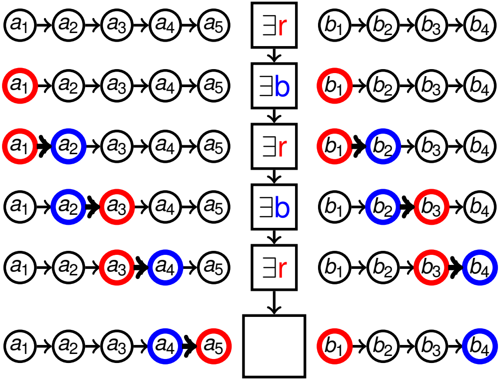
$QVT^2(L_5, L_4)$



$QVT^2(L_5, L_4)$

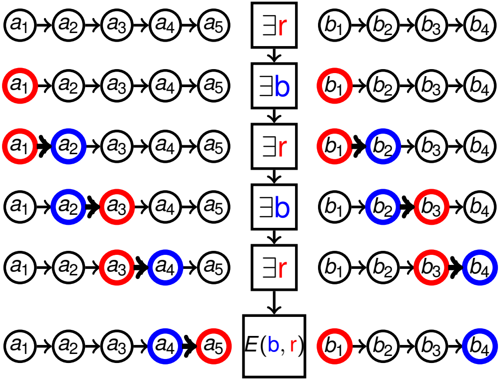


$QVT^2(L_5, L_4)$



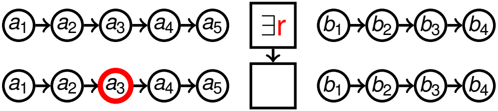
$QVT^2(L_5, L_4)$

Spoiler wins $QVT^2_5(L_5, L_4)$; Can he do better?

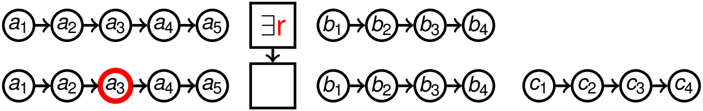


Spoiler wins $QVT_5^2(L_5, L_4)$;   Duplicator wins $QVT_4^2(L_5, L_4)$.

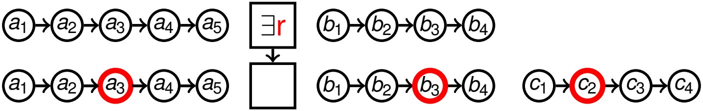
Spoiler wins $QVT_5^2(L_5, L_4)$; Duplicator wins $QVT_4^2(L_5, L_4)$.



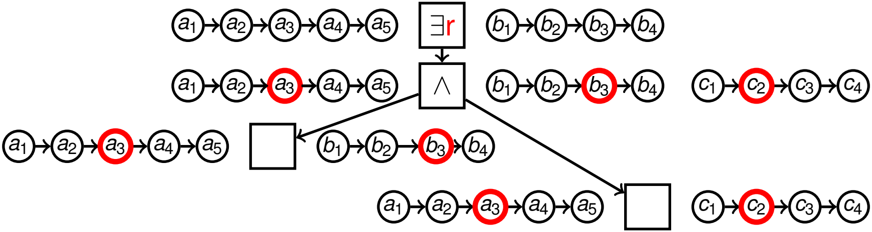
Spoiler wins $QVT^2_5(L_5, L_4)$; Duplicator wins $QVT^2_4(L_5, L_4)$.

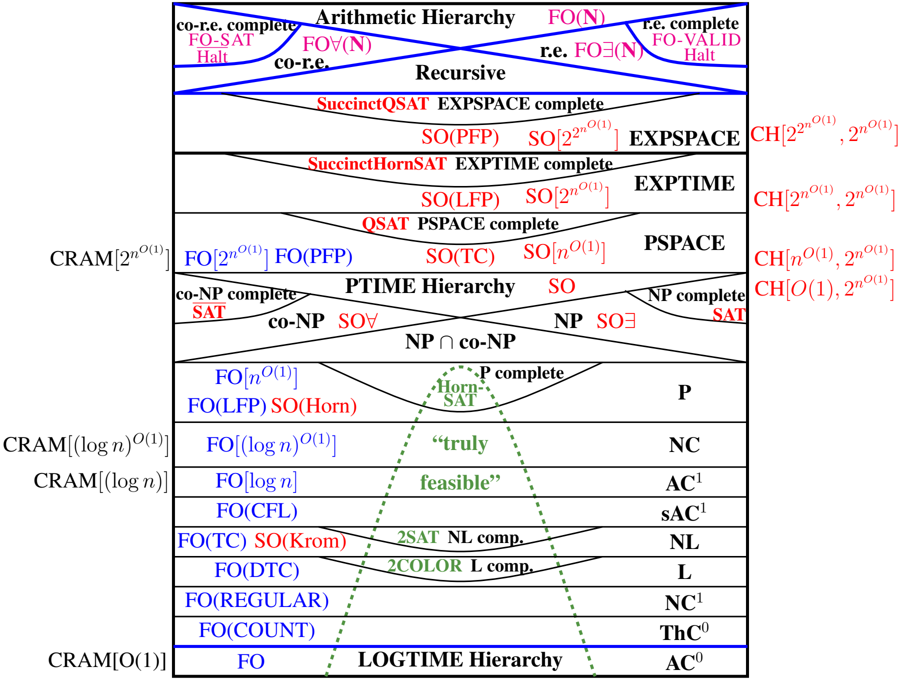


Spoiler wins $QVT_5^2(L_5, L_4)$;   Duplicator wins $QVT_4^2(L_5, L_4)$.



Spoiler wins $QVT_5^2(L_5, L_4)$;   Duplicator wins $QVT_4^2(L_5, L_4)$.





Arithmetic Hierarchy FO(**N**)

co-r.e. complete FO-SAT Halt

FO∀(**N**) **co-r.e.**

r.e. complete FO-VALID Halt

r.e. FO∃(**N**)

Recursive

SuccinctQSAT **EXPSPACE complete**

SO(PFP) SO$[2^{2^{n^{O(1)}}}]$ **EXPSPACE** CH$[2^{2^{n^{O(1)}}}, 2^{n^{O(1)}}]$

SuccinctHornSAT **EXPTIME complete**

EXPTIME

SO(LFP) SO$[2^{n^{O(1)}}]$ CH$[2^{n^{O(1)}}, 2^{n^{O(1)}}]$

QSAT **PSPACE complete**

CRAM$[2^{n^{O(1)}}]$ FO$[2^{n^{O(1)}}]$ FO(PFP) SO(TC) SO$[n^{O(1)}]$ **PSPACE** CH$[n^{O(1)}, 2^{n^{O(1)}}]$

PTIME Hierarchy SO CH$[O(1), 2^{n^{O(1)}}]$

co-NP complete SAT

NP complete SAT

co-NP SO∀ **NP** SO∃

NP ∩ co-NP

FO$[n^{O(1)}]$ **P complete**

Horn-SAT

FO(LFP) SO(Horn) **P**

CRAM$[(\log n)^{O(1)}]$ FO$[(\log n)^{O(1)}]$ **“truly** **NC**

CRAM$[(\log n)]$ FO$[\log n]$ **feasible”** **AC^1**

FO(CFL) **sAC^1**

FO(TC) SO(Krom) **2SAT** **NL comp.** **NL**

FO(DTC) **2COLOR** **L comp.** **L**

FO(REGULAR) **NC^1**

FO(COUNT) **ThC^0**

CRAM[O(1)] FO **LOGTIME Hierarchy** **AC^0**

Conclusions and Future Directions

- Let's learn to play these games better, especially the *QVT* game: for fun and improving our knowledge.

Conclusions and Future Directions

- Let’s learn to play these games better, especially the *QVT* game: for fun and improving our knowledge.
- Let’s look again at some great previous lower bounds including the following, among others, and try hard to reprove them and extend them using the *QVT* game:

Conclusions and Future Directions

- Let's learn to play these games better, especially the *QVT* game: for fun and improving our knowledge.
- Let's look again at some great previous lower bounds including the following, among others, and try hard to reprove them and extend them using the *QVT* game:
  - Grohe Schweikardt: Succinctness [GS05]

Conclusions and Future Directions

- Let's learn to play these games better, especially the *QVT* game: for fun and improving our knowledge.
- Let's look again at some great previous lower bounds including the following, among others, and try hard to reprove them and extend them using the *QVT* game:
    - Grohe Schweikardt: Succinctness [GS05]
    - Rossman: Tight Variable Hierarchy [R08]

Conclusions and Future Directions

- Let's learn to play these games better, especially the *QVT* game: for fun and improving our knowledge.
- Let's look again at some great previous lower bounds including the following, among others, and try hard to reprove them and extend them using the *QVT* game:
  - Grohe Schweikardt: Succinctness [GS05]
  - Rossman: Tight Variable Hierarchy [R08]
  - Hella Väänänen: Formula Size [HV15]

Conclusions and Future Directions

- Let’s learn to play these games better, especially the *QVT* game: for fun and improving our knowledge.
- Let’s look again at some great previous lower bounds including the following, among others, and try hard to reprove them and extend them using the *QVT* game:
  - Grohe Schweikardt: Succinctness [GS05]
  - Rossman: Tight Variable Hierarchy [R08]
  - Hella Väänänen: Formula Size [HV15]
- I’ll be here the whole weekend; come say, “Hello”; let’s talk about these and related issues.

Conclusions and Future Directions

- Let's learn to play these games better, especially the *QVT* game: for fun and improving our knowledge.
- Let's look again at some great previous lower bounds including the following, among others, and try hard to reprove them and extend them using the *QVT* game:
  - Grohe Schweikardt: Succinctness [GS05]
  - Rossman: Tight Variable Hierarchy [R08]
  - Hella Väänänen: Formula Size [HV15]
- I'll be here the whole weekend; come say, "Hello"; let's talk about these and related issues.
- Thank you!